リレーショナルデータベースの基礎

■ リレーションスキーマ

この資料では，リレーションスキーマは「リレーション名(属性名リスト)」であるとする．例えば，リレーション T が，表 1 の通りであるとする．このとき，リレーション T のリレーションスキーマは T(name, score, student_name)である．

表 1 リレーション T の例

name	score	student_name
Database	80	KK
Database	95	AA
Database	80	LL
Programming	85	KK
Programming	75	LL

■ リレーショナルデータモデル

リレーショナルデータモデルでは，データベースを，リレーションの集まりとして記述する．一貫性制約の記述では，主キー制約，一意制約，非空制約，ドメイン制約，参照整合性制約などの所定の制約を記述する．

■ リレーショナルデータモデルでのデータ操作の体系

リレーショナルデータモデルでのデータ操作の体系は，リレーショナル代数演算およびリレーショナル関係論理を基礎とする．

■ リレーショナル代数

リレーショナル代数は，集合論に基礎を置く体系である．集合演算と，リレーショナル代数に特有の演算がある．リレーショナル代数とリレーショナル関係論理は，土台となる数理が違うものの，相互に等価であるので，リレーショナル代数のみを説明する．

【集合演算】

和集合，差集合，共通集合，直積集合

【リレーショナル代数に特有の演算】

射影，選択，結合，商

■ 和両立

リレーション $R(A_1, A_2, \cdots, A_n)$ と $S(B_1, B_2, \cdots, B_m)$ が和両立であるとは，次の 2 条件を満足しているときを言う．

(1) R と S の次数が等しい．つまり $n = m$

(2) 各 i ($1 \leqq i \leqq$ n) について，A_i と B_i のドメインが等しい．

■ 和集合演算

R と S を和両立なリレーションであるとする．R と S の和集合演算は「$R\cup S$」と書く．定義は次の通りである．

$$R\cup S=\{\,t\mid t\in R\ \vee\ t\in S\}$$

■ 差集合演算

R と S を和両立なリレーションであるとする．R と S の差集合演算は「$R-S$」と書く．定義は次の通りである．

$$R-S=\{\,t\mid t\in R\ \wedge\ \neg(\,t\in S\,)\}$$

■ 共通集合演算

R と S を和両立なリレーションとする．R と S の共通集合演算は「$R\cap S$」と書く．定義は次の通りである．

$$R\cap S=\{\,t\mid t\in R\ \wedge\ t\in S\}$$

■ 直積集合演算

$R(A_1, A_2, \cdots, A_n)$ と $S(B_1, B_2, \cdots, B_m)$をリレーションとする，R と S の直積集合演算は「$R\times S$」と書く．定義は次の通りである．

$$R\times S=\{\,(r, s)\mid r\in R\ \wedge\ s\in S\}$$

但し，「(r, s)」と書いているのは，$r=(a_1, a_2, \cdots, a_n)$, $s=(b_1, b_2, \cdots, b_m)$ とするときに，$(r, s)=(a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_m)$ なる $n+m$ 項のタップルである．

■ 射影演算

$R(A_1, A_2, \cdots, A_n)$ をリレーションとする．全属性集合 $\{A_1, A_2, \cdots, A_n\}$ の部分集合 $X=\{A_{i1}, A_{i2}, \cdots, A_{ik}\}$について，$R$ の X上の射影は，「$R[X]$」あるいは「$R[A_{i1}, A_{i2}, \cdots, A_{ik}]$」と書く．定義は次の通りである．

$$R[A_{i1}, A_{i2}, \cdots, A_{ik}]=\{\,u\mid u\in dom(A_{i1})\times dom(A_{i2})\times\cdots\times dom(A_{ik})$$
$$\wedge\ (\exists t\in R)\ (t[A_{i1}]=u[A_{i1}]\wedge t[A_{i2}]=u[A_{i2}]\wedge\ \cdots\ \wedge t[A_{ik}]=u[A_{ik}])\}$$

※「$R[A_{i1}, A_{i2}, \cdots, A_{ik}]$」の代わりに「$\pi_{A_{i1}, A_{i2}, \cdots, A_{ik}}(R)$」のように書くこともある．

表 2(a)のリレーション T に対する選択 T[score] は，表 2(b) に示すようなリレーションになる．

表 2 射影の例

(a) リレーション T の例（表 1 の再掲）

name	score	student_name
Database	80	KK
Database	95	AA
Database	80	LL
Programming	85	KK
Programming	75	LL

(b) (a)のリレーション T に対する射影 T[score]

score
80
95
80
85
75

■ 選択演算

$R(A_1, A_2, \cdots, A_n)$ をリレーションとする．R の属性 A_i と A_j 上の θ-選択演算を「$R[A_i\ \theta\ A_j]$」と書く．定義は次の通りである．θ は 2 項述語である．

$$R[A_i\ \theta\ A_j] = \{\ t \mid t \in R \wedge t[A_i]\ \theta\ t[A_j]\ \}$$

R の属性 A_i と値 c に関する θ-選択演算を「$R[A_i\ \theta\ c]$」と書く．定義は次の通りである．

$$R[A_i\ \theta\ c] = \{\ t \mid t \in R \wedge t[A_i]\ \theta\ c\}$$

表 2-Y(a)のリレーション T に対する選択 T[score] は，表 2-Y(b) に示すようなリレーションになる．

表 3 選択の例

表 1 のリレーション T に対する選択 T[score > 85]

name	score	student_name
Database	95	AA
Programming	85	KK

■ リレーショナル代数での比較演算子

比較演算子は 2 項述語である．リレーショナル代数では，数値及び文字列に関する比較演算子θとして，$\theta \in \{>, \geqq, =, \leqq, <, \neq\}$ の 6 種類を考えるのが普通である．

※ SQL では「<=」，「>=」，「<>」のように書く．SQL と混同しないこと．

■ θ -結合演算

$R(A_1, A_2, \cdots, A_n)$ と $S(B_1, B_2, \cdots, B_m)$をリレーションとする．R の属性 A_i と S の属性 B_j 上の θ -結合演算を「$R[A_i\ \theta\ B_j]S$」と書く．定義は次の通りである．θ は比較演算子であり，2 項演算であることに注意せよ．

$R[Ai\ \theta\ Bj]$S は，R と S の直積集合（$R \times S$）の中から「$R.Ai\ \theta\ S.Bj$」を満足する要素を選んだもの

θ -結合演算は，直積演算と，θ-選択演算を使って，次のように定義することもできる．

$$\boxed{R[A_i\ \theta\ B_j]S = (R \times S)\ [R.A_i\ \theta\ S.B_j]}$$

リレーショナル代数式を使って次のように書くことができる．

$$\boxed{R[A_i\ \theta\ B_j]S = \{\ (t, u) \mid t \in R \land u \in S \land t[A_i]\ \theta\ u[B_j]\ \}}$$

但し，「(t, u)」と書いているのは，$t = (a_1, a_2, \cdots, a_n)$, $u = (b_1, b_2, \cdots, b_m)$ とするときに，$(t, u) = (a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_m)$ なる $n + m$ 項のタップルである．

※「$R[A_i\ \theta\ B_j]S$」の代わりに「$R \bowtie_{A_i\ \theta\ B_j} S$」のような書き方をすることもある．

※　ここでは，R の単一の属性 A_i と S の単一の属性 B_j の結合演算の説明にとどめる．複数の属性での結合演算については，説明を割愛する．

■ 結合属性

$R(A_1, A_2, \cdots, A_n)$ と $S(B_1, B_2, \cdots, B_m)$をリレーションとする．この 2 つのリレーションの共通属性 $C_1, C_2, \cdots, C_k$ は次の性質を満たす．

(1) $\{A_1, A_2, \cdots, A_n\} \cap \{B_1, B_2, \cdots, B_m\} = \{C_1, C_2, \cdots, C_k\}$

(2) 各 C_i $(1 \leqq i \leqq k)$ について，元のリレーション R と S におけるドメインが等しい．つまり $\mathrm{dom}(R.C_i) = \mathrm{dom}(S.C_i)$

<u>共通属性のことを結合属性ともいう</u>.

■ 自然結合演算

$R(A_1, A_2, \cdots, A_n)$ と $S(B_1, B_2, \cdots, B_m)$をリレーションとする．<u>**R と S の自然結合演算を「$R * S$」と書く**</u>．自然結合演算は，結合演算と，射影演算を使って，次のように表すことができる．

$R * S = (R[C_1 = C_1, C_2 = C_2, \cdots, C_k = C_k]S)\ [A_1, A_2, \cdots, A_n, D_1, D_2, \cdots, D_{m\text{-}k}]$,

但し，$C_1, C_2, \cdots, C_k$ は，A と B の結合属性.

また，$\{B_1, B_2, \cdots, B_m\} - \{C_1, C_2, \cdots, C_k\} = \{D_1, D_2, \cdots, D_{m\text{-}k}\}$とする．

※「$R * S$」の代わりに「$R \bowtie S$」のように書くこともある．

■ 商演算

$R(A_1, A_2, \cdots, A_{n\text{-}m}, B_1, B_2, \cdots, B_m)$ を n 次のリレーション，$S(B_1, B_2, \cdots, B_m)$を m 次のリレーションとする（但し，$m < n$）．R を S で割った商を「$R \div S$」と書く．定義は次の通りである．

$R \div S = \{\ t \mid t \in R[A_1, A_2, \cdots, A_{n\text{-}m}] \land (^\forall u \in S)\ ((t, u) \in R)\}$

但し，「(t, u)」と書いているのは，$t = (a_1, a_2, \cdots, a_{n\text{-}m})$, $u = (b_1, b_2, \cdots, b_m)$ とするときに，$(t, u) = (a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_m)$ なる n 項のタップルである.「$^\forall$」は，全称作用素であり，全て (for all) を表す.

「$R[A_1, A_2, \cdots, A_{n\text{-}m}]$」と書いているのは射影演算であり，「$\{\ u \mid u \in dom(A_1) \times dom(A_2) \times \cdots \times dom(A_{n\text{-}m}) \land (^\exists t \in R)\ (t[A_1] = u[A_1] \land t[A_2] = u[A_2] \land \cdots \land t[A_{n\text{-}m}] = u[A_{n\text{-}m}])\}$」のことである．

表 4. リレーショナル代数のまとめ

	記法	定義 ※　定義では論理式などを使用．論理式の記法は，集合を基礎とする論理式にしている．
和集合演算	$R \cup S$	$R \cup S = \{\, t \mid t \in R \vee t \in S \}$
差集合演算	$R - S$	$R - S = \{\, t \mid t \in R \wedge \neg(t \in S) \}$
共通集合演算	$R \cap S$	$R \cap S = \{\, t \mid t \in R \wedge t \in S \}$
直積集合演算	$R \times S$	$R \times S = \{ (r, s) \mid r \in R \wedge s \in S \}$ ※ (r, s) は，$r = (a_1, a_2, \cdots, a_n)$, $s = (b_1, b_2, \cdots, b_m)$ とするときに，$(r, s) = (a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_m)$ なる $n + m$ 項のタップル
射影演算	$R[A_{i1}, A_{i2}, \cdots, A_{ik}]$	$R[A_{i1}, A_{i2}, \cdots, A_{ik}] = \{ u \mid u \in dom(A_{i1}) \times dom(A_{i2}) \times \cdots \times dom(A_{ik})$ $\wedge (^{\exists} t \in R) (t[A_{i1}] = u[A_{i1}] \wedge t[A_{i2}] = u[A_{i2}] \wedge \cdots \wedge t[A_{ik}] = u[A_{ik}])\}$
選択演算	$R[A_i \,\theta\, A_j]$	$R[A_i \,\theta\, A_j] = \{\, t \mid t \in R \wedge t[A_i] \,\theta\, t[A_j] \}$
	$R[A_i \,\theta\, c]$	$R[A_i \,\theta\, c] = \{\, t \mid t \in R \wedge t[A_i] \,\theta\, c \}$
結合演算	一般の結合演算 $R[A_i \,\theta\, B_j]S$	$R[A_i \,\theta\, B_j]S = \{ (t, u) \mid t \in R \wedge u \in S \wedge t[A_i] \,\theta\, u[B_j] \}$ ※ (t, u) は，$t = (a_1, a_2, \cdots, a_n)$, $u = (b_1, b_2, \cdots, b_m)$ とするときに，$(t, u) = (a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_m)$ なる $n + m$ 項のタップル
	自然結合 $R * S$	$R * S = (R[C_1 = C_1, C_2 = C_2, \cdots, C_k = C_k]S)\,[A_1, A_2, \cdots, A_n, D_1, D_2, \cdots, D_{m-k}]$, ※ $C_1, C_2, \cdots, C_k$ は，A と B の結合属性， $\{B_1, B_2, \cdots, B_m\} - \{C_1, C_2, \cdots, C_k\} = \{D_1, D_2, \cdots, D_{m-k}\}$
商演算	$R \div S$	$R \div S = \{\, t \mid t \in R[A_1, A_2, \cdots, A_{n-m}] \wedge (^{\forall} u \in S)\,((t, u) \in R)\}$

上記の 8 つの演算は必ずしも独立ではない．例えば，共通集合は，差集合を使って $R \cap S = R - (R - S)$ のようになる．結合は，直積と選択を表すことができる．商は，直積と射影と差を使って表すことができる．

■　リレーションの分解（decomposition）

あるリレーションが，リレーションスキーマ $R(A_1, A_2, \cdots, A_n)$ のインスタンスであるとする．このリレーションの，m 個のリレーション $X_1, X_2, \cdots, X_m$ への分解は次のように定義される．

$X_1, X_2, \cdots, X_m$ を R の全属性集合$\{A_1, A_2, \cdots, A_n\}$の部分集合で，

$$X_1 \cup X_2 \cup \cdots \cup X_m = \{A_1, A_2, \cdots, A_n\}$$

なる条件を満たすとき（X_i と X_j は共通集合をもってよい），このリレーションを m 個の射影 $R[X_1]$, $R[X_2]$, $\cdots$, $R[X_m]$で置き換えることを，このリレーションの分解という．各 $R[X_i]$を R の分解成分という．

■ 関数従属性

リレーションスキーマ $R(A_1, A_2, \cdots, A_l, B_1, B_2, \cdots, B_m, C_1, C_2, \cdots, C_n)$ に関数従属性 $A_1, A_2, \cdots, A_l \rightarrow B_1, B_2, \cdots, B_m$ が存在するとは，次の場合をいう．

リレーションスキーマ R の任意のインスタンスの任意の 2 タップル t と t' について，それらの $A_1, A_2, \cdots, A_l$ 値が等しければ，$B_1, B_2, \cdots, B_m$ 値が等しいということが必ず成り立つ．

■ 属性名の記法

複数のリレーションで属性名が同じことがありえるため，属性名 A と書いただけでは，どのリレーションであるかのどの属性であるかを特定できない場合がある．特定のリレーション T の特定の属性名 A を特定したい場合, リレーション名 T と属性名 A を連結して T.A のように書く場合がある．

■ **べき集合の例**

集合{みかん，りんご，バナナ}から構成されるべき集合は，{ ϕ , {みかん}, {りんご}, {バナナ}, {みかん，りんご}, {りんご，バナナ}, {バナナ，みかん}, {みかん，りんご，バナナ}}である．

■ **第一正規形**

ドメインとして，分解不可能な「単純値」のみを対象とすること．

■ SQL 列制約（column-constraint）

SQL 列制約の代表的なものを下に列挙する．

PRIMARY KEY ・・・ 主キー制約
NOT NULL ・・・ 非空制約
UNIQUE ・・・ 一意制約
CHECK (<expression>) ・・・ 更新時にチェックされる式（expression）
FOREIGN KEY (<column-name>) REFERENCES <foreign-table> (<column-name>, ...) ・・・ 参照整合性制約（つまり外部キー）